

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 06012045 A

(43) Date of publication of application: 21.01.1994

(51) Int. Cl. G09G 5/24

B41J 2/485, B41J 5/30, G06F 3/12, G06F 15/66, G09G 5/30

(21) Application number: 04167927

(22) Date of filing: 25.06.1992

(71) Applicant: RICOH CO LTD

(72) Inventor: NIRO MASAKAZU

(54) CHARACTER IMAGE ROTATION
PROCESSING METHOD FOR IMAGE
FORMING DEVICE

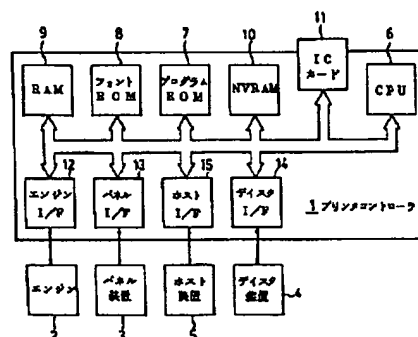
(57) Abstract:

PURPOSE: To shorten the time required for rotation processing of character images and to reduce the memory by adding information indicating whether each character image holds the symmetrical form at the time of rotation or not to the character image and referring to this information to use the character image, which holds the symmetrical form then, without rotation as it is.

CONSTITUTION: When a character code is received from a host device 5, it is discriminated whether rotation is instructed or not; and if it is not instructed, a character image corresponding to this code is taken out from a font ROM 8 and is expanded in a scan buffer of a RAM 9. If it is instructed, a griff descriptor in font data of a font ROM 8 is referred to discriminate whether the character image corresponding to the re-

ceived character code holds the symmetrical form at the time of rotation, and the character image is expanded in the scan buffer of the RAM 9 if it holds the symmetrical form. If it does not hold the symmetrical form, the griff descriptor is referred to check the presence/absence of an alternative character image, and the alternative character image is expanded in the scan buffer in the case of the presence.

COPYRIGHT: (C)1994,JPO&Japio

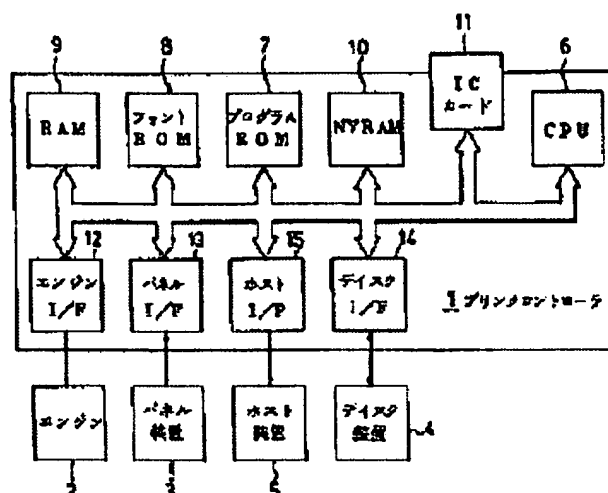


JP6012045 - CHARACTER IMAGE ROTATION PROCESSING METHOD FOR IMAGE FORMING DEVICE

Biblio info

Abstract:

Source: JP6012045A2 PURPOSE: To shorten the time required for rotation processing of character images and to reduce the memory by adding information indicating whether each character image holds the symmetrical form at the time of rotation or not to the character image and referring to this information to use the character image, which holds the symmetrical form then, without rotation as it is. CONSTITUTION: When a character code is received from a host device 5, it is discriminated whether rotation is instructed or not; and if it is not instructed, a character image corresponding to this code is taken out from a font ROM 8 and is expanded in a scan buffer of a RAM 9. If it is instructed, a griff descriptor in font data of a font ROM 8 is referred to discriminate whether the character image corresponding to the received character code holds the symmetrical form at the time of rotation, and the character image is expanded in the scan buffer of the RAM 9 if it holds the symmetrical form. If it does not hold the symmetrical form, the griff descriptor is referred to check the presence/absence of an alternative character image, and the alternative character image is expanded in the scan buffer in the case of the presence.



Claims of JP6012045:

Machine_translation_Claims:

Claim 1 In the image formation equipment which takes out the character image corresponding to reception and this code for a character code, and creates image information This information is referred to, when each character image is made to rotate each of that image, the information which shows whether a symmetry form is held is added to it, respectively and a revolution is specified as it. It is the character image revolution art characterized by using the character image which the character image which does not hold a symmetry form rotates when it is made to rotate, and holds a symmetry form as it is, without making it rotate.

Claim 2 In the image formation equipment which takes out the character image corresponding to reception and this code for a character code, and creates image information When the character image at the time of making it rotate becomes equal to the character image which others do not rotate The information on the character code which carries out this alternative activity with the information which shows that the alphabetic character which can carry out an

alternative activity exists in the character image of the origin of it is added. The character image revolution art characterized by using the image of this alternative-command character, without rotating the original character image when the alphabetic character which can carry out an alternative activity with reference to this information when a revolution is specified exists.

Description of JP6012045:

Machine_translation Detailed Description of the Invention:

0001

Industrial Application This invention relates to the character image revolution art in image formation equipments, such as printer equipments, such as a laser beam printer, a digital copier, and a regular paper FAX.

0002

Description of the Prior Art When it has only a character image for portraits (columnar writing) from the former and the land CAPE (lateral writing: 90-degree revolution) and 180 revolutions are specified, he is trying to rotate all the character images to be used in image formation equipments, such as a laser beam printer.

0003

Problem(s) to be Solved by the Invention However, since what (a symmetry form is held) does not change even if it rotates in which direction like "O" in a character image, the thing which becomes equal to the original character image when it rotates 180 degrees for the business of "- " were contained, when all the character images to be used were rotated, while the useless processing time started, the useless memory for storing it was needed. This invention is made in view of the above-mentioned point, and while shortening the time amount which revolution processing of a character image takes, it aims at reducing the amount of the memory used.

0004

Means for Solving the Problem In the image formation equipment which takes out the character image corresponding to reception and this code for a character code, and creates image information in order that this invention may attain the above-mentioned object This information is referred to, when each character image is made to rotate each of that image, the information which shows whether a symmetry form is held is added to it, respectively and a revolution is specified as it. When it is made to rotate, the character image which does not hold a symmetry form makes it rotate, and the character image holding a symmetry form offers the character image revolution art used as it is, without making it rotate.

0005 moreover, when the character image at the time of making it rotate becomes equal to the character image which others do not rotate The information on the character code which carries out this alternative activity with the information which shows that the alphabetic character which can carry out an alternative activity exists in the character image of the origin of it is added. When the alphabetic character which can carry out an alternative activity with reference to this information when a revolution is specified exists, the character image revolution art which uses the image of this alternative-command character, without rotating the original character image is also offered.

0006

Function The information is referred to, when according to the character image revolution art of the 1st invention rotating each of that image, and the information which shows whether a symmetry form is held is added to each character image, respectively and a revolution is specified as it. Since it is used as it is, without the character image which does not hold a symmetry form making it rotate, and the character image holding a symmetry form making it rotate when it is made to rotate, while the time amount which revolution processing of a character image takes is shortened, the amount of the memory used decreases.

0007 when the character image at the time of making it rotate becomes equal to the character image which others do not rotate according to the character image revolution art of the 2nd invention (for example, it becomes equal to "" (" -- the character image at the time of making it rotate 180 degrees -- ")) The information on the character code which carries out an alternative activity with the information which shows that the alphabetic character which can carry out an alternative activity exists in the character image of the origin of it is added. Since the image of an alternative-command character is used without rotating the original character image when the alphabetic character which can carry out an alternative activity with reference to the information when a revolution is specified exists, the same effectiveness as **** can be acquired.

0008 In addition, by combining each above-mentioned character image revolution art, the time amount which revolution processing of a character image takes can be shortened further, and considerable reduction also of the amount of the memory used can be carried out.

0009

Example Hereafter, the example of this invention is concretely explained based on a drawing. Drawing 2 is the block block diagram of a laser beam printer showing one example of this invention, and consists of a printer controller 1, and an engine 2, panel equipment 3 and a disk unit 4. Host equipment 5 is high order equipment of this printer.

0010 a printer controller 1 -- every of CPU6, a program ROM 7, fonts ROM8, RAM9, and NVRAM10, IC card 11, the engine interface (an interface is called I/F for short below) 12 and panel I/F13, disk I/F14, and host I/F15 -- it consists of I/F.

0011 CPU6 is a central processing unit which controls the printer controller 1 whole by the mode directions from the program and the panel equipment 3 of a program ROM 7, and the command from host equipment (host computer) 5. The read only memory in which the program ROM 7 stores the control program of this printer controller 1, and a font ROM 8 are read only memories which store two or more font data with which a typeface differs from magnitude etc.

0012 RAM9 is a random access memory used for the font file for storing the scanning buffer (bit map buffer) for storing the page buffer for storing the input buffer for storing the work-piece memory for CPU6, and input data, and page data, and an image image data (bit map data), and a download font etc. NVRAM10 is nonvolatile memory which memorizes the content of the mode directions from panel equipment 3 etc., and IC card 11 is a removable memory card used when supplying font data and a program from the outside.

0013 An interface for engine I/F12 to perform the communication link of the engine 2, the command and the status which print actually, or printing data, and panel I/F13 are interfaces which perform the communication link of the panel equipment 3 which tells a user about the condition of a current printer, or performs mode directions, and the command and the status. Disk I/F14 is an interface for communicating with a disk unit 4.

0014 Host I/F15 is host equipment 5 and an interface which performs a communication link, and is usually SENTORO I/F and RS232C. A disk unit 4 is external storage which memorizes data with font data, various programs, printing data, etc., and are a floppy disk drive unit (FDD), a hard disk drive unit (HDD), etc.

0015 Usually, the character image of the portrait most often used is beforehand stored in the font ROM 8. The example of a configuration of the font data stored in the font ROM 8 at drawing 3 is shown. Font header which shows what kind of thing the font is at the head of font data (Font header) It is. There, the information which shows of what kind of typeface they are the magnitude of a font and a font is stored.

0016 The information in the degree is a glyph table (Glyph table). It is shown where the character image corresponding to the character code called and inputted is stored. Moreover, it is also possible to get to know that there is no character image corresponding to a character code by the value. The information on each character image is stored next. The information on each character image consists of glyph images (Glyph image) which are the glyph descriptor (Glyph descriptor) which stores the information about a character image, and a actual image data.

0017 CPU6 in a printer controller 1 chooses one font data by the command from host equipment 5 from two or more font data stored in the font ROM 8, when a character code is received, it chooses a character image out of it, and it develops it to the scanning buffer secured to RAM9.

0018 If there is a rotational command from host equipment 5 at this time, after choosing the character image corresponding to a character code, that character image will be developed to a scanning buffer, after performing revolution processing etc. Once storing the character image which processed as font data on RAM9 as arts, such as a revolution, there are an approach of developing on the scanning buffer of same RAM9 and the approach of processing, whenever it receives a character code, and developing to a scanning buffer.

0019 Although a memory area will be used for an excess since font data must be stored also in memory areas other than the scanning buffer of RAM9 in the case of the former, since processing becomes unnecessary in using the once processed character image again, the processing time can be shortened. Whenever it receives also by the same character image in the case of the latter, it is necessary to process but, and since it is not necessary to store font data in any memory areas other than a scanning buffer, a memory area can be saved.

0020 however, a thing which does not change an image at all into a character image even if it rotates in which direction like "O" -- ". -- " -- although a printing location changes like, the image itself has some not changing. In the former, these alphabetic characters were also processing the revolution etc. For this reason, memory was consumed vainly and the useless processing time had started.

0021 The example of a configuration of the conventional glyph descriptor is shown for the example of a configuration of the glyph descriptor in the font data of drawing 3 (information on each character image) in a table 2 at a table 1, respectively. Moreover, a glyph type (glyph type) example of a table 1 is shown in drawing 4 , and a glyph type (glyph type) example of a table 2 is shown in drawing 5 , respectively.

0022

A table 1

0023

A table 2

0024 In this example, the information which shows whether a symmetry form is held for each of that image to each character image, respectively 90 degrees and when rotating 180 degrees is added so that they may be seen and understood. Moreover, when data serve as variable length and the character image at the time of making it rotate becomes equal to the character image which others do not rotate, the character code of the information which shows that the character image which can carry out an alternative activity at the time of a revolution exists in the original character image, and an alternative-command character is added.

0025 Drawing 1 is a flow chart which shows the processing concerning this invention by CPU6 of drawing 2. First, when a character code is received from host equipment 5, it judges whether the revolution was specified or not and the revolution is not specified, the character image corresponding to the received character code is taken out from a font ROM 8, and it develops to the scanning buffer of RAM9.

0026 Moreover, when a revolution is specified, the glyph descriptor in the font data stored in the font ROM 8 is referred to. It judges whether a symmetry form is held when the character image corresponding to the received character code rotates it. In holding, it develops the character image as it is to the scanning buffer of RAM9, and in not holding, it check and it is whether there is any alternative-command character image or there is nothing, and , with reference to a glyph descriptor, it will develop the alternative-command character image to the scanning buffer of RAM9.

0027 When return to the first step, the same processing as **** is repeated, it ends and the image image data for 1 page is generated if it judges whether expansion of the character image for 1 page was completed next and has not ended after the expansion to the scanning buffer of one character image is completed, the image image data is transmitted to an engine 2, and carries out a printed output.

0028 For example, the printed output of the image image data to which each character image rotated 90 degrees of 90-degree revolution commands to the carrier beam case must be carried out from host equipment 5. Usually, since many character images do not hold a symmetry form when rotating 90 degrees, they need to perform revolution processing. Even if however, "O" rotates 180 degrees, it holds 90 degrees of symmetry forms. b05, b04, b03, and b02 of the glyph type shown in drawing 4 of the glyph descriptor of this "O" are set to "1", respectively.

0029 By it, it can know that CPU6 is a symmetrical alphabetic character even if 90 degrees "O" rotates by referring to the glyph descriptor of "O" when the character code of "O" is received. Therefore, the processing time is shortened by developing as it is, without making a scanning buffer rotate the character image of "O."
 0030 On the other hand, "***" does not hold a symmetry form, 90 degrees or when rotating 180 degrees, but since it becomes equal to "***" which others do not rotate when rotating 90 degrees, "***" can be used instead. Moreover, "***" can be instead used for a case the bottom rotated 180 degrees. Although graph type b03 and b02 of this alphabetic character are reset by "0", it turns out that b09, b07, and b06 are set to "1", and there is an alternative-command character image. Moreover, the character code of "***" and "***" is added to the usual glyph descriptor.

0031 Therefore, the glyph type to a character image will know not holding a symmetry form, when it rotates, and that an alternative-command character image can be used, and will pull out the character code of "***" from glyph disk PURITA, and CPU6 will develop the character image of "***" to a scanning buffer instead of the character image which rotated "***", if the character code of "***" is received.

0032 As mentioned above, although the example which applied this invention to the laser beam printer was explained, of course, this invention can be applied to other optical printers, such as not only this but an LED printer, a liquid crystal shutter printer, etc., at various kinds of image formation equipments, such as a digital copier and a regular paper FAX.

0033

Effect of the Invention As explained above, according to the character image revolution art of this invention, the time amount which revolution processing of a character image takes can be shortened, and, moreover, the amount of the memory used can be reduced.

Brief Description of the Drawings

Drawing 1 It is flow drawing showing the processing concerning this invention by CPU6 of drawing 2 .

Drawing 2 It is the block block diagram of a laser beam printer showing one example of this invention.

Drawing 3 It is memory map drawing showing the example of a configuration of the font data stored in the font ROM 8 of drawing 1 .

Drawing 4 It is the explanatory view showing a glyph type example of the glyph descriptor in the font data of drawing 3 .

Drawing 5 It is the explanatory view showing a glyph type example of the glyph descriptor in the conventional font data.

Description of Notations

1 Printer Controller 2 Engine

3 Panel Equipment 4 Disk Unit

5 Host Equipment 6 Central Processing Unit (CPU)

7 Program ROM 8 Font ROM

9 RAM 10 NVRAM

Drawing 5

Drawing 1

Drawing 3

Drawing 2

Drawing 4

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-12045

(43) 公開日 平成6年(1994)1月21日

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 9 G 5/24		9061-5G		
B 4 1 J 2/485				
5/30	F	8907-2C		
G 0 6 F 3/12	H	8804-2C	B 4 1 J 3/12	T

審査請求 未請求 請求項の数 2 (全 7 頁) 最終頁に続く

(21) 出願番号 特願平4-167927

(22) 出願日 平成4年(1992)6月25日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 丹路 雅一

東京都大田区中馬込1丁目3番6号 株式会社リコー内

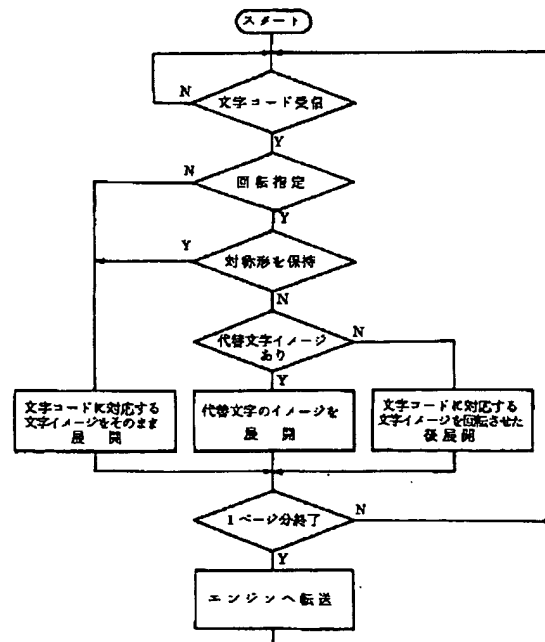
(74) 代理人 弁理士 大澤 敬

(54) 【発明の名称】 画像形成装置における文字イメージ回転処理方法

(57) 【要約】

【目的】 文字イメージの回転処理に要する時間を短縮すると共に、メモリの使用量を低減する。

【構成】 各文字イメージに、その各イメージを回転させたときに対称形を保持するか否かを示す情報をそれぞれ付加し、回転が指定された場合にその情報を参照して、回転させたときに対称形を保持しない文字イメージは回転させ、対称形を保持する文字イメージは回転させずにそのまま使用する。また、回転させたときの文字イメージが他の回転させない文字イメージと等しくなる場合には、その元の文字イメージに代替使用できる文字が存在することを示す情報と代替使用する文字コードの情報を付加し、回転が指定された場合に、その情報を参照して代替使用できる文字が存在する場合には元の文字イメージを回転させずに代替文字のイメージを使用する。



【特許請求の範囲】

【請求項1】 文字コードを受け取り、該コードに対応する文字イメージを取り出して画像情報を作成する画像形成装置において、

各文字イメージに、その各イメージを回転させたときに対称形を保持するか否かを示す情報をそれぞれ付加し、回転が指定された場合に該情報を参照して、回転させたときに対称形を保持しない文字イメージは回転させ、対称形を保持する文字イメージは回転させずにそのまま使用することを特徴とする文字イメージ回転処理方法。

【請求項2】 文字コードを受け取り、該コードに対応する文字イメージを取り出して画像情報を作成する画像形成装置において、

回転させたときの文字イメージが他の回転させない文字イメージと等しくなる場合には、その元の文字イメージに代替使用できる文字が存在することを示す情報と該代替使用する文字コードの情報を付加し、回転が指定された場合に、該情報を参照して代替使用できる文字が存在する場合には元の文字イメージを回転させずに該代替文字のイメージを使用することを特徴とする文字イメージ回転処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、レーザプリンタ等のプリンタ装置、デジタル複写機、普通紙FAX等の画像形成装置における文字イメージ回転処理方法に関する。

【0002】

【従来の技術】 レーザプリンタ等の画像形成装置においては、従来からポートレイト（縦書き）用の文字イメージのみを持ち、ランドスケープ（横書き：90度回転）や180度の回転が指定された場合には、使用する全ての文字イメージを回転させるようにしている。

【0003】

【発明が解決しようとする課題】 しかしながら、文字イメージの中には「O」のようにどの方向に回転しても変わらない（対称形を保持する）ものや、「-」の用に180度回転した場合には元の文字イメージと等しくなるものなどが含まれているため、使用する全ての文字イメージを回転させると、無駄な処理時間がかかると同時にそれを格納するための無駄なメモリを必要としていた。この発明は上記の点に鑑みてなされたものであり、文字イメージの回転処理に要する時間を短縮すると共に、メモリの使用量を低減することを目的とする。

【0004】

【課題を解決するための手段】 この発明は、上記の目的を達成するため、文字コードを受け取り、該コードに対応する文字イメージを取り出して画像情報を作成する画像形成装置において、各文字イメージに、その各イメージを回転させたときに対称形を保持するか否かを示す情報をそれぞれ付加し、回転が指定された場合に該情報を

参照して、回転させたときに対称形を保持しない文字イメージは回転させ、対称形を保持する文字イメージは回転させずにそのまま使用する文字イメージ回転処理方法を提供する。

【0005】 また、回転させたときの文字イメージが他の回転させない文字イメージと等しくなる場合には、その元の文字イメージに代替使用できる文字が存在することを示す情報と該代替使用する文字コードの情報を付加し、回転が指定された場合に、該情報を参照して代替使用できる文字が存在する場合には元の文字イメージを回転させずに該代替文字のイメージを使用する文字イメージ回転処理方法も提供する。

【0006】

【作用】 第1の発明の文字イメージ回転処理方法によれば、各文字イメージに、その各イメージを回転させたときに対称形を保持するか否かを示す情報をそれぞれ付加し、回転が指定された場合にその情報を参照して、回転させたときに対称形を保持しない文字イメージは回転させ、対称形を保持する文字イメージは回転させずにそのまま使用するので、文字イメージの回転処理に要する時間が短縮すると共に、メモリの使用量が低減する。

【0007】 第2の発明の文字イメージ回転処理方法によれば、回転させたときの文字イメージが他の回転させない文字イメージと等しくなる場合（例えば「I」を180度回転させたときの文字イメージは「I」と等しくなる）には、その元の文字イメージに代替使用できる文字が存在することを示す情報と代替使用する文字コードの情報を付加し、回転が指定された場合に、その情報を参照して代替使用できる文字が存在する場合には元の文字イメージを回転させずに代替文字のイメージを使用するので、上述と同様の効果を得られる。

【0008】 なお、上記の各文字イメージ回転処理方法を組み合わせることにより、文字イメージの回転処理に要する時間をさらに短縮させ、メモリの使用量も相当低減させることができる。

【0009】

【実施例】 以下、この発明の実施例を図面に基づいて具体的に説明する。図2はこの発明の一実施例を示すレーザプリンタのブロック構成図であり、プリンタコントローラ1とエンジン2、パネル装置3、ディスク装置4とからなる。ホスト装置5は、このプリンタの上位装置である。

【0010】 プリンタコントローラ1は、CPU6、プログラムROM7、フォントROM8、RAM9、NV-RAM10、ICカード11と、エンジンインタフェース（以下インタフェースをI/Fと略称する）12、パネルI/F13、ディスクI/F14、ホストI/F15の各I/Fから構成されている。

【0011】 CPU6は、プログラムROM7のプログラムとパネル装置3からのモード指示、及びホスト装置

(ホストコンピュータ) 5からのコマンドによってプリンタコントローラ1全体を制御する中央処理装置である。プログラムROM7は、このプリンタコントローラ1の制御プログラムを格納しているリードオンリ・メモリ、フォントROM8は、書体や大きさなどが異なる複数のフォントデータを格納しているリードオンリ・メモリである。

【0012】RAM9は、CPU6用のワークメモリ、入力データを格納するためのインプットバッファ、ページデータを格納するためのページバッファ、画像イメージデータ(ビットマップデータ)を格納するためのスキャンバッファ(ビットマップバッファ)、ダウンロードフォントを格納するためのフォントファイル等に使用するランダムアクセス・メモリである。NVRAM10は、パネル装置3からのモード指示の内容などを記憶しておく不揮発性メモリであり、ICカード11は、フォントデータやプログラムを外部から供給する場合に使用する着脱可能なメモリカードである。

【0013】エンジンI/F12は、実際に印字を行なうエンジン2とコマンド及びステータスや印字データの通信を行なうためのインタフェース、パネルI/F13は、使用者に現在のプリンタの状態を知らせたりモード指示を行なったりするパネル装置3と、コマンドやステータスの通信を行なうインタフェースである。ディスクI/F14は、ディスク装置4と通信を行なうためのインタフェースである。

【0014】ホストI/F15は、ホスト装置5と通信を行なうインタフェースであり、通常はセントロI/FやRS232Cである。ディスク装置4は、フォントデータやプログラム、印字データなどの様々のデータを記憶しておく外部記憶装置であり、フロッピディスク装置(FDD)やハードディスク装置(HDD)などである。

【0015】通常、もっともよく使用されるポートレイトの文字イメージは予めフォントROM8に格納されている。図3に、フォントROM8に格納されているフォントデータの構成例を示す。フォントデータの先頭にはそのフォントがどのようなものであるかを示すフォントヘッダ(Font header)がある。そこには、フォントの大きさやどのような書体のフォントであるかを示す情報等が格納されている。

【0016】その次にある情報は、グリフテーブル(Glyph table)と呼ばれ、入力された文字コードに対応する文字イメージが何処に格納されているのかを示している。また、その値により文字コードに対応する文字イメ

ージがないことを知ることも可能である。この後に、各文字イメージの情報が格納される。各文字イメージの情報は、文字イメージに関する情報を格納しているグリフディスクリプタ(Glyph descriptor)と実際のイメージデータであるグリフイメージ(Glyph image)から構成される。

【0017】プリンタコントローラ1内のCPU6は、ホスト装置5からの指令により、フォントROM8に格納されている複数のフォントデータの中から1つのフォントデータを選択し、文字コードを受け取った時にその中から対応する文字イメージを選択して、RAM9に確保したスキャンバッファに展開する。

【0018】このとき、ホスト装置5から回転などの指令があると、文字コードに対応する文字イメージを選択した後、その文字イメージを回転処理などを行なった上でスキャンバッファに展開する。回転などの処理方法としては、処理を行なった文字イメージをRAM9上に一旦フォントデータとして格納した後、同じRAM9のスキャンバッファ上に展開する方法と、文字コードを受け取るたびに処理を行なってスキャンバッファに展開する方法とがある。

【0019】前者の場合は、RAM9のスキャンバッファ以外のメモリ領域にもフォントデータを格納しなければならないためメモリ領域を余分に使用することになるが、一旦処理した文字イメージを再度使用する場合には処理が不要になるため、処理時間を短縮させることができる。後者の場合は、同じ文字イメージでも受けとるたびに処理をする必要があるが、スキャンバッファ以外のメモリ領域にフォントデータを格納する必要がないため、メモリ領域を節約することができる。

【0020】しかし、文字イメージの中には、「○」のようにどの方向に回転してもイメージが全く変わらないようなものや、「。」の様に印字位置は変わるがイメージそのものは変わらないものもある。従来では、これらの文字も回転等の処理を行なっていた。このため、メモリを無駄に消費したり、無駄な処理時間がかかったりしていた。

【0021】表1に図3のフォントデータ中のグリフディスクリプタ(各文字イメージの情報)の構成例を、表2に従来のグリフディスクリプタの構成例をそれぞれ示す。また、図4に表1のグリフタイプ(glyph type)の一例を、図5に表2のグリフタイプ(glyph type)の一例をそれぞれ示す。

【0022】

【表1】

項 目	データの型	内 容
image_off	long	グリフディスクリプタの先頭からグリフイメージまでのオフセット
glyph_type	u_short	グリフに関する情報
left_off	short	アクティブポジションからグリフイメージの左端までの距離
top_off	short	アクティブポジションからグリフイメージの上端までの距離
image_width	u_short	実際のイメージの幅(ドット数)
image_height	u_short	実際のイメージの高さ(ドット数)
cell_width	u_short	文字のセル幅(ドット数)
set_width	long	文字の移動量
char_coda_1	u_long	代替文字の文字コード1
char_coda_2	u_long	代替文字の文字コード2

【0023】

20【表2】

項 目	データの型	内 容
image_off	long	グリフディスクリプタの先頭からグリフイメージまでのオフセット
glyph_type	u_short	グリフに関する情報
left_off	short	アクティブポジションからグリフイメージの左端までの距離
top_off	short	アクティブポジションからグリフイメージの上端までの距離
image_width	u_short	実際のイメージの幅(ドット数)
image_height	u_short	実際のイメージの高さ(ドット数)
cell_width	u_short	文字のセル幅(ドット数)
set_width	long	文字の移動量

【0024】それらを見て分かるように、この実施例においては、各文字イメージに、その各イメージを90°及び180°回転させたときにそれぞれ対称形を保持するか否かを示す情報が付加されている。また、データは可変長となり、回転させたときの文字イメージが他の回

【0025】図1は、図2のCPU6によるこの発明に係わる処理を示すフローチャートである。まず、ホスト装置5から文字コードを受け取った時に、回転が指定されたか否かを判断し、回転が指定されていない場合には、受け取った文字コードに対応する文字イメージをフォントROM8から取り出してRAM9のスキャンバッ

ファに展開する。

【0026】また、回転が指定された場合には、フォントROM8に格納されているフォントデータ中のグリフディスクリプタを参照して、受け取った文字コードに対応する文字イメージがそれを回転させたときに対称形を保持するか否かを判断し、保持する場合にはその文字イメージをRAM9のスキャンバッファにそのまま展開し、保持しない場合にはグリフディスクリプタを参照して、代替文字イメージがあるかないかをチェックし、あればその代替文字イメージをRAM9のスキャンバッファに展開する。

【0027】1つの文字イメージのスキャンバッファへの展開が終了すると、次に1ページ分の文字イメージの展開が終了したか否かを判断して、終了していなければ最初のステップに戻って上述と同様の処理を繰り返し、

終了して1ページ分の画像イメージデータが生成された場合には、その画像イメージデータをエンジン2へ転送してプリント出力させる。

【0028】例えば、ホスト装置5から90°の回転指令を受けた場合には、各文字イメージが90°回転された画像イメージデータがプリント出力されなければならない。通常、多くの文字イメージは90°回転させたときに対称形を保持しないため、回転処理を施す必要がある。しかし、例えば「○」は90°、180°回転させても対称形を保持する。この「○」のグリフディスクリプタの図4に示すグリフタイプのb05、b04、b03、b02はそれぞれ「1」にセットされている。

【0029】それによって、CPU6は「○」の文字コードを受け取った時に、「○」のグリフディスクリプタを参照することにより、「○」は90°回転しても対称な文字であることを知ることができる。したがって、「○」の文字イメージをスキャンバッファに回転させずにそのまま展開することにより、処理時間が短縮する。

【0030】一方、「⊥」は90°又は180°回転させたときには対称形を保持しないが、90°回転させたときに他の回転させない「⊥」と等しくなるため、「⊥」を代わりに使用することができる。また、180°回転させた場合には「⊥」を代わりに使用することができる。この文字のグラフィックタイプのb03、b02は「0」にリセットされているが、b09、b07、b06が「1」にセットされており、代替文字イメージがあることが分かる。また、通常のグリフディスクリプタには「⊥」と「⊥」の文字コードが付加されている。

【0031】したがって、CPU6は「⊥」の文字コードを受け取ると、そのグリフタイプから、文字イメージは回転したときに対称形を保持しないことと、代替文字イメージを使用することができることを知り、グリフディスクリプタから「⊥」の文字コードを引き出し、

「⊥」を回転させた文字イメージの代わりに「⊥」の文字イメージをスキャンバッファに展開する。

【0032】以上、この発明をレーザプリンタに適用した実施例について説明したが、この発明はこれに限らず、LEDプリンタ、液晶シャッタープリンタ等の他の光プリンタには勿論、デジタル複写機、普通紙FAX等の各種の画像形成装置に適用し得るものである。

【0033】

【発明の効果】以上説明してきたように、この発明の文字イメージ回転処理方法によれば、文字イメージの回転処理に要する時間を短縮し、しかもメモリの使用量を低減することができる。

【図面の簡単な説明】

【図1】図2のCPU6によるこの発明に係わる処理を示すフロー図である。

【図2】この発明の一実施例を示すレーザプリンタのブロック構成図である。

【図3】図1のフォントROM8に格納されているフォントデータの構成例を示すメモリマップ図である。

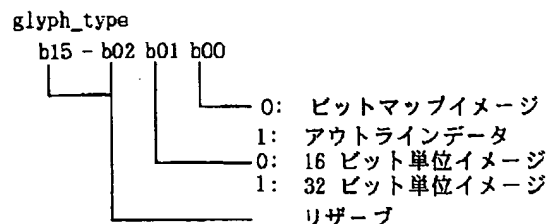
【図4】図3のフォントデータ中のグリフディスクリプタのグリフタイプの一例を示す説明図である。

【図5】従来のフォントデータ中のグリフディスクリプタのグリフタイプの一例を示す説明図である。

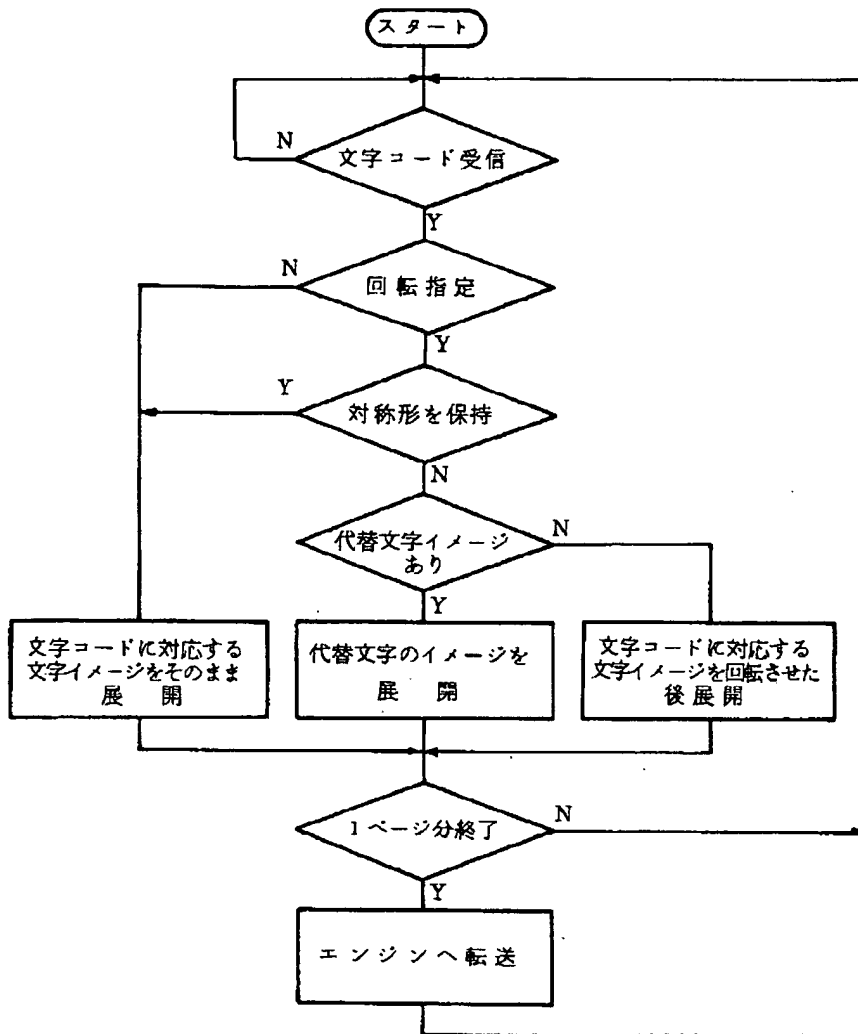
【符号の説明】

1 プリンタコントローラ	2 エンジン
3 パネル装置	4 ディスク装置
5 ホスト装置	6 中央処理装置(CPU)
7 プログラムROM	8 フォントROM
9 RAM	10 NVRAM

【図5】



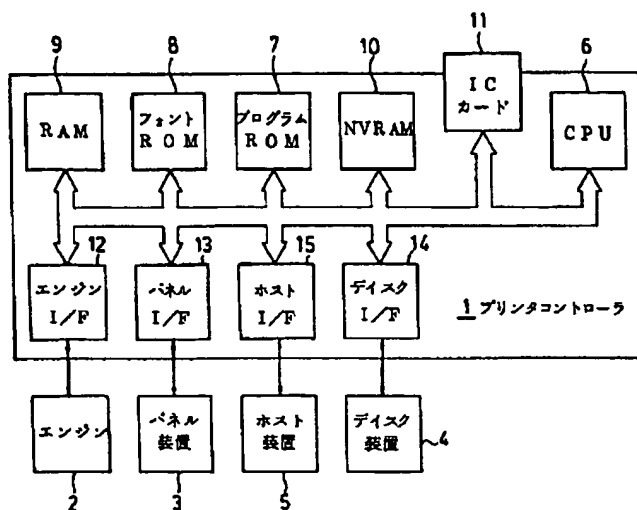
【図1】



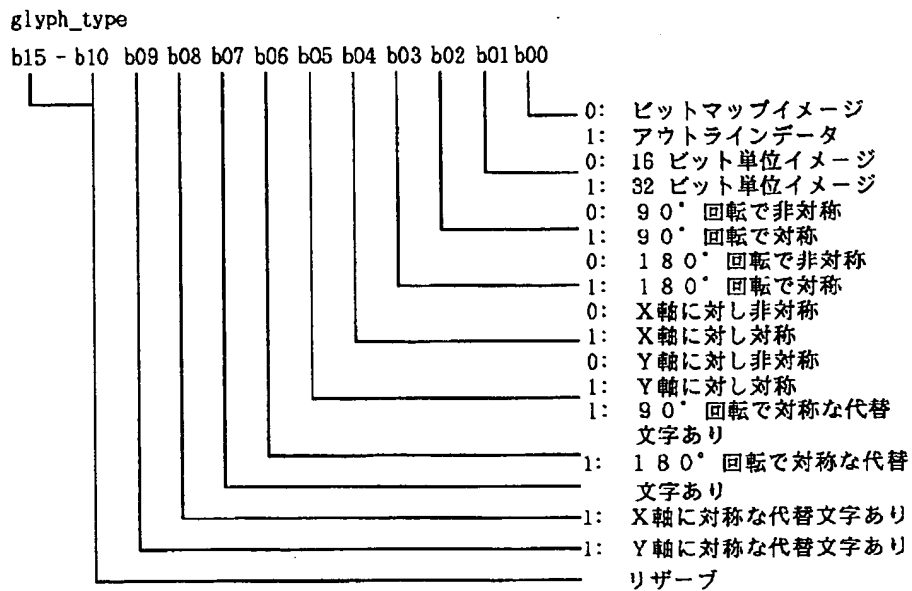
【図3】

フロントヘッダ
グリフテーブル
グリフディスクリプタ
グリフイメージ
グリフディスクリプタ
グリフイメージ
グリフディスクリプタ
グリフイメージ
グリフディスクリプタ
グリフイメージ

【図2】



【図4】



フロントページの続き

(51) Int. Cl.⁵

G 0 6 F 15/66

G 0 9 G 5/30

識別記号

3 5 0

庁内整理番号

8420-5L

9061-5G

F I

技術表示箇所